International Journal of Engineering and Applied Computer Science (IJEACS)

# Features for Detecting Malware on Computing Environments

Ajit Kumar, K.S. Kuppusamy
Department of Computer Science
Pondicherry University
Puducherry-605014, India

G. Aghila
Department of Computer Science and Engineering
NIT Puducherry
Karaikal-609609, India

***Abstract*—Malware is the main threat for all computing environments. It also acts as launching platform for many other cyber threats. Traditional malware detection system is not able to detect "modern", "unknown" and "zero-day" malware. Recent developments in computing hardware and machine learning techniques have emerged as alternative solution for malware detection. The efficiency of any machine learning algorithm depends on the features extracted from the dataset. Various types of features are extracted and being researched with machine learning approach to detect malware that are targeted towards computing environments. In this work we have organized and summarized different feature types used to detect malware. This work will direct future researchers and industry to make decision on feature type selection regarding chosen computing environment for building an accurate malware classifier.**

***Keywords- malware; computing environment; cyber-threat; feature type; machine learning***

## I. INTRODUCTION

Malware is a computer program, intensely written to harm computing resources. Malware are of different type based on structural and behaviors difference, such as Virus, Worm, Trojan, Bot, Spyware, Adware, Rootkit, Bootkit, Ransomware etc. Growth of variant of known malware and new malware is increasing year-by-year [1] and posing threat to digital infrastructure.

Malware is main threat for all four kind of computing environments: 1) Personal computing; 2) Mobile computing; 3) Embedded computing; and 4) Industrial control system (ICS) computing. Although a large percentage of total malware targeted for first two environments i.e. personal and mobile computing, recent past have seen a major surge in malware targeting other two environments as well i.e. embedded and ICS computing. To tackle malware threats personal and mobile computing have some traditional solutions such as signature and heuristic based anti-malware but embedded and ICS computing are wide open for malware attack. Traditional solutions are not effective in detecting malware at either of computing environments as they have inherited limitations [2]–[4].

Signature based techniques are backbone of these traditional anti-malware solutions which itself is totally unable to work against *"modern", "unknown" and "zero-day" malware*. Signature based techniques works in two phases: 1) Signature creation; and 2) Signature matching. Signature creation is a multi-step process which involves steps such as malware collection, malware analysis, signature generation and signature distribution. All of these steps are carry out with the help of human and machine in proportion. Human involvement makes process costly and slow which provide a large attack window to malware. Machine works on the principal of generalization which misses artifacts of "modern", "unknown" and "zero-day" malware. Signature matching is also mutli-step process such as file scanning, signature look-up and alerting user. It performance is dependent on previous phase i.e. signature creation because it can only match signatures which are in the database and so "zero-day" and "unknown" malware will escape the detection. Apart from technical bottlenecks, signature based techniques also suffers with other drawbacks like costly analysis process, high computation and memory requirements at end host and requirement of regular signature updates. These bottlenecks and drawbacks of signature based techniques created a need of alternative anti-malware solution and machine learning based techniques is emerging to fulfill the same.

Machine learning techniques consider malware detection as a binary or multi-class classification problem which is similar to many other domains. Machine learning based malware detection has two phases: 1) training; and 2) classification. Training is a multi-step process and sequential in nature. Steps involve in building malware classifier are: sample collection, sample labeling, feature extraction, feature selection and model building. Sample collection is process of collecting malware and benign programs which is precedence by sample labeling which assign true class label (malware and benign) to each sample. Labeled samples are ready for further step which is feature extraction. Feature extraction is very important step of overall machine learning process. Extracted features mainly decide classifier performance and so with different type of features classifier perform differently. This decisive nature of feature attracts lots of engineering methods to extract different


*ISBN: 978-0-9957075-1-1*

type of features which have more discriminative values than others. Feature selection deals with excessive extracted features and help to filter out only few useful features on the basis of discriminative rank. Model building is last step of training, it takes selected features and run machine learning algorithms which output a model which would be able to classify inputted new sample. During classification phase each sample goes through the feature extraction phase but now only those features are extracted on which the model is built. Built model takes these extracted features as input and output the probable class label for each sample.

Machine learning based malware detection is suitable for "modern", "unknown" and "zero-day" malware detection because it is not per sample base technique as signature based techniques are, instead it works by learning malware and benign classification based on extracted features from training dataset and able to generalized to unseen samples. Features type plays an important and decisive role for accurate malware detection using machine learning. Many researches in domain of malware detection using machine learning focus on different feature type which are extracted by various methods and impact classifier performance.

Over years many feature type for malware detection is proposed and experimented which are spread over all of computing environments. In this work, we have organized and summarized different feature type used for various file types on different computing environments. Due to vary computing architecture, supported file types and analysis method among different computing environments, feature types vary across these environments. Classifiers performance depend largely on features types and feature selection, hence having a well organized literature on feature types will help in easy decision making for various entities involved such as future researchers and industries developers.

## II. METHODS

Feature type can be group primarily according to computing environments and further under each computing environment it can be organized according to the analysis type. In this section four computing environments are explained and then each of analysis type is explained.

### A. *Computing Environments*

Computing environment is term use to describe a complete computing platform comprise of hardware, operating system (OS), and other software. Each computing environments differs on aforementioned components. Each of computing environment is explained further.

*1)Personal computing:* Personal computing refer to the use of Desktop and laptop computer which are used in normal day-to-day life and in various enterprises to automate the tasks. Distinguished dimensions of Personal computing environment are a bigger output screen, high internal and external memory, desktop OS and attached keyboard and mouse.

*2)Mobile computing:* Mobile computing refer to all of those devices which are mobile in nature and having a smaller screen size than personal computing devices and limited with small battery. All such devices have specific mobile operating system and customized operating system. Android, iOS and Windows are there main leading mobile OS.

*3)Embedded computing:* Embedded computing refers to all those smart digital devices which have configuration options and run a specialized operating system designed for such embedded system. OS running in smart car, smart home, modern freeze and many other modern digital appliances are example of embedded computing.

*4)ICS computing:* Industrial control system (ICS) computing refers to those devices which run specialized OS and software to control and monitor industrial system such as digital power and water distribution system, nuclear plant etc.

### B. *Analysis and Feature Type*

Feature extraction involves two type of analysis which carried out in different manners and gives features which have vary discriminative values. Static and dynamic are two type of malware analysis techniques which provide three different types of features: 1) Static features; 2) Dynamic features; and 3) Hybrid features. Each of three feature type is explained further.

*1)Static features:* Static analysis is method of malware analysis, in which sample are analyzed statically i.e. without executing the sample and only structural and physical property are analyze. Static feature are those features which are extracted by aforementioned static analysis method. Static analysis are safe and fast because sample are not executed hence the analysis platform will not be affected and so many samples can be analyzed without cleaning the analysis environment. Static feature are easy to extract without it doesn't not require complex execution and monitoring process.

*2)Dynamic features:* Dynamic analysis is process of executing the sample, monitoring the analysis environment and recording the changes made during the execution time. Dynamic features are those features which are extracted from the recorded changes of dynamic analysis. Dynamic analysis is time consuming and complex but it handles many of the limitations of static analysis such as it enable to extract features from packed and obfuscated malware sample which can't handle by static analysis.

*3)Hybrid features:* Hybrid features are combination of static and dynamic features. Integrating static and dynamic features enrich the discriminative power of feature set and improve the performance of the malware classifier. Although it's very beneficial but same time it is very costly in term of analysis time and computing.


ISBN: 978-0-9957075-1-1

## III. Features for Personal Computing

Personal computing has a larger user base because of it use in various domains. Windows OS is leading with respect to number of users. Large number of users attracts the attackers which results in huge number of malware targeting only Windows user. Similarly, detection solution is also centric toward Windows malware. In this section different features are listed and explain which are used to build malware classifier.

### A. Strings

Every executable or any other files have "strings" in it source which have been used as feature for building malware classifiers. All strings present in source files are extracted by static or dynamic methods and used as features following text classification approach. Static method for strings extraction has explained in [5], [6] while strings collected during "runtime trace" (by dynamic analysis) have been explained and used in [7].

### B. DLL & API Call

DLL and API call are also used as features for malware detection. These two can be extracted by both static and dynamic analysis method. DLL and API are used as Boolean features, which is prepared by extracting all DLL and API call from malware and benign class and taken as features. Present and absent of DLL and API use in a sample is considered as '1' and '0' respectively [8], [9].

### C. Byte-n-grams

Byte-n-grams use the frequency of "n" consecutive bytes in hexadecimal representation of a given sample as feature. Byte-n-grams are achieved by static analysis in two steps: 1) Converting sample to its hexadecimal representation, and 2) Processing and extracting byte-n-grams. Byte-n-grams based feature set is frequently used to build malware classifier [5], [10]–[14].

### D. Opcode-n-grams

Opcode-n-grams use the frequency of "n" consecutive opcode in assembly representation of a given sample as feature. Opcode-n-grams are achieved by static analysis in two steps: 1) Disassemble the sample, and 2) Processing and extracting opcode-n-grams. Opcode-n-grams based features have two variants, one with consider operand along with opcode and other which doesn't take operand in consideration [12], [15]–[20].

### E. PE Header Fields

PE headers fields values are also used as feature set for building malware classifier. This feature is not applicable to all file types but limited to all PE file format such DLL, exe etc. DOS_HEADER, FILE_HEADER and OPTIONAL_HEADER are three main headers from which fields value are extracted and used as feature. Various approach existed to utilize these values but all of them are carried out by static analysis method. With variation classification performance vary [8], [9], [21]–[27].

### F. Network and Host Activity

Dynamic analysis provides way to monitored and extracted dynamic behaviors such as network and host activities. During the analysis time every interaction of network and host is monitored and recorded. From recorded files various kind of features are extracted and used for building machine learning based malware classifier [28]–[30].

### G. Image Properties

Image properties are being used as features in image classification domain but by converting binary files to image can tap this potential for malware classification. With this motivation Nataraj et al. have converted binary file to grayscale image and then extracted GIST features from image to build malware classification system [31].

### H. Hardware Features

Hardware activity is bottom of any program execution and so monitoring it gives an accurate representation of program behaviors. Wang et al. [32] have used hardware interaction based features for building malware detection system.

## IV. Features for Mobile Computing

Android is leading operating system for mobile computing which spread across smart-phone to tablet. Due to a larger user base, Android is main target of security attacks and malware is one of major threat to Android. Many research works have considered this challenge and have proposed many solutions to keep Android safe from malware attacks. To the limitations of signature based detection, most of current works are focused on machine learning based Android malware detection. In this section, different features which are devised and used to build android malware detection system are listed and a short description along with appropriate work is presented.

### A. Permissions and Intents

Permissions and Intents are very crucial for any Android application; it decides the app functionality and behaviors. Using permissions and intents as features resulted in high classification performance for Android malware. These two are mostly used as Boolean features but few works have considered these as numeric feature [33]–[41]. Presence and absence of permissions and intents are taken as Boolean features whereas assigned weight of each permission is taken in numeric features.

### B. Strings

Similar to desktop files, Android applications also have different type of stings to accomplish various tasks. By extracting and using these strings, a feature set can be build to classify Android application into malware and benign [42].

### C. System Calls

System calls are bridge between user space and kernal space, a user event results into one or more system calls. By recording system call patterns of malware and benign, a


ISBN: 978-0-9957075-1-1

Boolean feature set can be created to build Android malware classification system [34], [43]–[47].

### D. Image Properties

Features extracted from image such as SIFT, GIST and HOG have demonstrated higher classification performance in computer vision domain. To utilize these features for Android malware classification, "apk to image" conversion is uses as per-processing step which convert *apk* file to image according to specified color map. From resulted image aforementioned features are extracted and used for building Android malware classifier [48].

### E. Network and Host Activities

Similar to personal computing mobile's network and host activities can be recorded while a sample is in execution. Such dynamic trace provides better representation of an application behaviors hence yields better classification accuracy with machine learning models [49]–[56].

TABLE I. VARIOUS FEATURES WITH THEIR ANALYSIS TYPE

| Features | Analysis Type |
|---|---|
| *Personal computing* | |
| Strings | Static & Dynamic |
| DLL & API call | Static |
| Byte-n-grams | Static |
| Opcode-n-grams | Static |
| PE headers fields | Static |
| Network and Host activities | Dynamic |
| Image properties | Static |
| Hardware features | Dynamic |
| *Mobile Computing* | |
| Permissions and Intents | Static |
| Strings | Static |
| System calls | Static & Dynamic |
| Image properties | Static |
| Network and Host activities | Dynamic |

## V. FEATURES FOR EMBEDDED COMPUTING

Embedded computing is getting popular due to improved hardware and advancement in software. Circuit based instruction methods are getting replaced with software alternatives which provides more functionality and flexibility (automated car, digital appliance etc.). This migration also brings threats associated with software such as malware. In recent past, few attacks on embedded system are reported but due to few users the malware problem is not getting attention. In future, with increase in user base and malware attacks this serious issue will be addressed.

## VI. FEATURES FOR ICS COMPUTING

Industrial Control System (ICS) comprise computing and network infrastructure for monitoring, automating and controlling the industrial system for example nuclear power plant, water and electric distribution & controlling network etc. Affect of attacking and damaging such infrastructure will be very dangerous and not only financial loss will occur but much life will be lost. *Stuxnet* [57] is one of such malware which was written and released targeting SCADA system installed in nuclear plant. Works have started to secure ICS system but use of machine learning is very limited. Most of works are focused on patching the known weakness of computer networks and systems. Solution targeting malware is very limited but future works will benefits of machine learning based solutions for securing ICS computing environment.

## VII. CONCLUSIONS AND FUTURE WORKS

In this work, we have summarized the different features used with machine learning to detect malware in various computing environment. This work provides a state-of-art status of machine learning based malware detection.

In future work, an empirical study will be conducted to validate the detection rate of various features and will try to filter out the most effective and efficient features to detect malware with respect to various computing environments.

## REFERENCES

[1] WebResource, "Av-Test.org," Online, 2016. [Online]. Available: https://www.av-test.org/en/statistics/malware/. [Accessed: 03-Dec-2016].

[2] P. Košinár, J. Malcho, and R. Marko, "Av testing exposed," no. September, pp. 1–7, 2010.

[3] D. Dagon and P. Vixie, "AV Evasion Through Malicious Generative Programs," pp. 1–6.

[4] S. Alvarez and T. Zoller, "The death of AV defense in depth?-revisiting anti-virus software," 2008.

[5] M. G. M. G. Schultz, E. Eskin, F. Zadok, and S. J. S. J. Stolfo, "Data mining methods for detection of new malicious executables," Proceedings 2001 IEEE Symposium on Security and Privacy, 2001, pp. 38–49.

[6] A. Shabtai, R. Moskovitch, Y. Elovici, and C. Glezer, "Detection of malicious code by applying machine learning classifiers on static features: A state-of-the-art survey," Inf. Secur. Tech. Rep., vol. 14, no. 1, pp. 16–29, Feb. 2009.

[7] K. Rieck, T. Holz, C. Willems, D. Patrick, P. Düssel, and P. Laskov, "Learning and classification of malware behavior," in Detection of Intrusions and Malware, and Vulnerability Assessment, Springer, 2008, pp. 108–125.

[8] M. Narouei, M. Ahmadi, G. Giacinto, H. Takabi, and A. Sami, "DLLMiner: Structural mining for malware detection," Secur. Commun. Networks, vol. 8, no. 18, pp. 3311–3322, 2015.


ISBN: 978-0-9957075-1-1

[9] T. Y. Wang, C. H. Wu, and C. C. Hsieh, "Detecting unknown malicious executables using portable executable headers," NCM 2009 - 5th Int. Jt. Conf. INC, IMS, IDC, pp. 278–284, 2009.

[10] N. Nissim, R. Moskovitch, L. Rokach, and Y. Elovici, "Novel active learning methods for enhanced PC malware detection in windows OS," Expert Syst. Appl., vol. 41, no. 13, pp. 5843–5857, Oct. 2014.

[11] R. K. Shahzad, S. I. Haider, N. Lavesson, and R. K. Shazhad, "Detection of spyware by mining executable files," in Availability, Reliability, and Security, 2010. ARES'10 International Conference on, 2010, pp. 295–302.

[12] A. Shabtai, R. Moskovitch, C. Feher, S. Dolev, and Y. Elovici, "Detecting unknown malicious code by applying classification techniques on OpCode patterns," Secur. Inform., vol. 1, no. 1, pp. 1–22, 2012.

[13] R. Moskovitch, C. Feher, N. Tzachar, E. Berger, M. Gitelman, S. Dolev, and Y. Elovici, "Unknown malcode detection using OPCODE representation," in Intelligence and Security Informatics, Springer, 2008, pp. 204–215.

[14] T. Abou-Assaleh, N. Cercone, V. Keselj, and R. Sweidan, "Detection of New Malicious Code Using N-grams Signatures.," PST, pp. 193–196, 2004.

[15] R. K. Shahzad, N. Lavesson, and H. Johnson, "Accurate Adware Detection Using Opcode Sequence Extraction," 2011 Sixth Int. Conf. Availability, Reliab. Secur., pp. 189–195, Aug. 2011.

[16] R. K. Shahzad and N. Lavesson, "Veto-based malware detection," in Availability, Reliability and Security (ARES), 2012 Seventh International Conference on, 2012, pp. 47–54.

[17] P. O'Kane, S. Sezer, K. McLaughlin, and E. G. Im, "SVM Training Phase Reduction Using Dataset Feature Filtering for Malware Detection," IEEE Trans. Inf. Forensics Secur., vol. 8, no. 3, pp. 500–509, Mar. 2013.

[18] M. E. Karim, A. Walenstein, A. Lakhotia, and L. Parida, "Malware phylogeny generation using permutations of code," J. Comput. Virol., vol. 1, no. 1–2, pp. 13–23, Sep. 2005.

[19] I. Santos, F. Brezo, B. Sanz, C. Laorden, and P. G. P. G. Bringas, "Using opcode sequences in single-class learning to detect unknown malware," IET Inf. Secur., vol. 5, no. 4, pp. 220–227, 2011.

[20] A. Lakhotia, A. Walenstein, C. Miles, and A. Singh, "VILO: a rapid learning nearest-neighbor classifier for malware triage," J. Comput. Virol. Hacking Tech., vol. 9, no. 3, pp. 1–15, Mar. 2013.

[21] M. Belaoued and S. Mazouzi, "A Real-Time PE-malware Detection System Based on CHI-Square Test and PE-File Features," in IFIP Advances in Information and Communication Technology, 2015, vol. 456, pp. 416–425.

[22] B. David, E. Filiol, and K. Gallienne, "Structural analysis of binary executable headers for malware detection optimization," J. Comput. Virol. Hacking Tech., pp. 1–7, 2016.

[23] Z. Markel and M. Bilzor, "Building a Machine Learning Classifier for Malware Detection."

[24] R. Merkel, T. Hoppe, C. Kraetzer, and J. Dittmann, "Statistical detection of malicious PE-executables for fast offline analysis," Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 6109 LNCS, pp. 93–105, 2010.

[25] A. Walenstein, D. J. Hefner, and J. Wichers, "Header information in malware families and impact on automated classifiers," Proc. 5th IEEE Int. Conf. Malicious Unwanted Software, Malware 2010, pp. 15–22, 2010.

[26] G. Yan, N. Brown, and D. Kong, "Exploring discriminatory features for automated malware classification," Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 7967 LNCS, pp. 41–61, 2013.

[27] J. H. Yang and Y. Ryu, "Toward an Efficient PE-Malware Detection Tool 1," vol. 109, pp. 14–17, 2015.

[28] U. Bayer, P. M. Comparetti, C. Hlauschek, C. Kruegel, and E. Kirda, "Scalable, Behavior-Based Malware Clustering.," in NDSS, 2009, vol. 9, pp. 8–11.

[29] C. Yavvari and A. Tokhtabayev, "Malware characterization using behavioral components," Comput. Netw. …, 2012.

[30] I. Gurrutxaga, O. Arbelaitz, J. M. Perez, J. Muguerza, J. I. Martin, and I. Perona, "Evaluation of malware clustering based on its dynamic behaviour," in Proceedings of the 7th Australasian Data Mining Conference-Volume 87, 2008, pp. 163–170.

[31] L. Nataraj, S. Karthikeyan, G. Jacob, and B. S. Manjunath, "Malware Images : Visualization and Automatic Classification," 2011.

[32] X. Wang, S. E. K. Chai, M. Isnardi, S. Lim, and R. Karri, "Hardware Performance Counter-Based Malware Identification and Detection with Adaptive Compressive Sensing," ACM Trans. Archit. Code Optim., vol. 13, no. 1, 2016.

[33] B. Sanz, I. Santos, C. Laorden, X. Ugarte-Pedrero, and P. G. Bringas, "On the automatic categorisation of android applications," 2012 IEEE Consum. Commun. Netw. Conf. CCNC'2012, pp. 149–153, 2012.

[34] P. P. K. Chan and W. K. Song, "Static detection of Android malware by using permissions and API calls," Proc. - Int. Conf. Mach. Learn. Cybern., vol. 1, pp. 82–87, 2015.

[35] S. Ju, H. Seo, and J. Kwak, "Research on android malware permission pattern using permission monitoring system," Multimed. Tools Appl., 2016.

[36] T. Nandhini and V. Arulmozhi, "Permission Tracking Security Model in Android Application," vol. 4, no. 2, pp. 6–12, 2015.

[37] F. Di Cerbo, A. Girardello, F. Michahelles, and S. Voronkova, "Detection of malicious applications on android OS," Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 6540 LNCS, pp. 138–149, 2011.

[38] H.-Y. Chuang and S.-D. Wang, "Machine Learning Based Hybrid Behavior Models for Android Malware Analysis," 2015 IEEE Int. Conf. Softw. Qual. Reliab. Secur., pp. 201–206, 2015.

[39] S. B. Almin and M. Chatterjee, "A novel approach to detect Android malware," Procedia Comput. Sci., vol. 45, no. C, pp. 407–417, 2015.

[40] M. Magdum, "Permission based Mobile Malware Detection System using Machine Learning Techniques," vol. 14, no. 6, pp. 6170–6174, 2015.

[41] S. Verma and S. K. Muttoo, "An Android Malware Detection Framework-based on Permissions and Intents," vol. 66, no. 6, pp. 618–623, 2016.

[42] A. MartIn, H. D.Menendez, and David Camacho, "String-based Malware Detection for Android Environments," in Intelligent Distributed Computing X, 2017, vol. 678, pp. 99–108.

[43] Dimjaševic, Marko, Simone Atzeni, Ivo Ugrina, and Zvonimir Rakamaric. "Android malware detection based on system calls." University of Utah, Tech. Rep (2015).

[44] M. Dimjaševi, S. Atzeni, and Z. Rakamari, "Evaluation of Android Malware Detection Based on System Calls," 2016.

[45] D. Arp, M. Spreitzenbarth, H. Malte, H. Gascon, and K. Rieck, "Drebin: Effective and Explainable Detection of Android Malware in Your Pocket," in Symposium on Network and Distributed System Security (NDSS), 2014, pp. 23–26.

[46] S. K. Dash, G. Suarez-tangil, S. Khan, K. Tam, M. Ahmadi, J. Kinder, and L. Cavallaro, "DroidScribe : Classifying Android Malware Based on Runtime Behavior," Mob. Secur. Technol., no. October, 2016.

[47] Y. Aafer, W. Du, and H. Yin, "DroidAPIMiner: Mining API-Level Features for Robust Malware Detection in Android," Secur. Priv. Commun. Networks, vol. 127, pp. 86–103, 2013.

[48] A. Kumar, K. P. Sagar, K. S. Kuppusamy, and G. Aghila, "Machine learning based malware classification for Android applications using multimodal image representations," 2016 10th Int. Conf. Intell. Syst. Control, pp. 1–6, 2016.

[49] M. K. Alzaylaee, S. Y. Yerima, and S. Sezer, "DynaLog : An automated dynamic analysis framework for characterizing Android applications," 2017.

[50] U. Zurutuza and N.-T. Simin, "Behavior-based malware detection system for the Android platform," 2011.

[51] M. Y. Wong and D. Lie, "IntelliDroid : A Targeted Input Generator for the Dynamic Analysis of Android Malware," no. February, pp. 21–24, 2016.

[52] A. Ali-gombe, I. Ahmed, and G. G. R. Iii, "AspectDroid : Android App Analysis System," pp. 145–147.

[53] H. Wang, Y. Guo, Z. Tang, G. Bai, and X. Chen, "Reevaluating Android Permission Gaps with Static and Dynamic Analysis," 2015.

[54] M. Spreitzenbarth, F. C. Freiling, F. Echtler, T. Schreck, and J. Hoffmann, "Mobile-Sandbox: Having a Deeper Look into Android Applications," ACM Symp. Appl. Comput., pp. 1808–1815, 2013.

[55] E. B. Siegfried Rasthofer, Steven Arzt, Marc Miltenberger, "Harvesting Runtime Data in Android Applications for Identifying Malware and Enhancing Code Analysis," Ndss, 2016.

[56] M. Zheng, M. Sun, and J. C. S. Lui, "DroidTrace : A Ptrace Based Android Dynamic Analysis System with Forward Execution Capability," pp. 1–6.

[57] R. Langner, "Stuxnet: Dissecting a cyberwarfare weapon," Secur. Privacy, IEEE, vol. 9, no. 3, pp. 49–51, 2011.

AUTHOR PROFILE

**Ajit Kumar** is a Ph.D. Research Scholar in the Department of Computer Science at Pondicherry Central University, Puducherry, India. He has completed his Master's degree in Computer Science in the year 2011 and Bachelor's degree in Computer Application in year 2009. His research interest includes Cyber Security, Malware Classification and Machine Learning. He has published 6 papers in International Conference realted to Malware and Machine Learning.

**K.S.Kuppusamy** is an Assistant Professor of Computer Science at Pondicherry Central University, India. He has received his Ph.D. in Computer Science and Engineering in the year 2013 and his master's degree in Computer Science and Information Technology in the year 2005. He has got a total of 11 years of teaching experience. His research interest includes Accessible Computing, Security and accessibility. He has published more than 25 papers in various international journals and coferences. He is the recipient of Best Teacher award during the years 2010, 2011 2013, 2015 and 2016.

**G. Aghila** is Professor at Department of Computer Science and Engineering, National Institute of Technology Puducherry, Karaikkal. She has got a total of 26 years of teaching experience. She has received her M.E (Computer Science and Engineering) and Ph.D. from Anna University, Chennai, India. She has published nearly 70 research papers in web crawlers and ontology based information retrieval. She has successfully guided 7 Ph.D. scholars. She was in receipt of Schrneiger award. She is an expert in ontology development. Her area of interest includes Intelligent Information Management, Artificial intelligence, Text mining and Semantic web technologies.




*ISBN: 978-0-9957075-1-1*